

CLAIM AMENDMENTS

Claim Amendment Summary

Claims pending

- Before this Amendment: Claims 1, 4-11, 13, 14, 17-26, 28, 32-41 and 43-47.
- After this Amendment: Claims 1, 4-11, 13, 14, 17-26, 28, 32-41 and 43-47.

Non-Elected, Canceled, or Withdrawn claims: None.

Amended claims: 1, 5, 13, 17, and 32.

New claims: None.

Claims:

1. (Currently Amended) A computer readable storage medium having a data programming model executable by a processor embodied thereon, the data programming model comprising:

a data item having a plurality of data item properties, each of these data item properties having an associated value;

a user interface element having an element property with a value that is defined by an association to a respective data item property;

a data style definition configured to define a visual representation of the data item on a user interface, wherein the associated values of the data item

properties influence the appearance of a plurality of features of the user interface element, wherein the data items are maintained independently from the data style definition, and wherein the data item is represented as a subtree of the user interface element and the data item properties are represented as part of the subtree;

a binding definition configured to associate the element property of the user interface element with the data item property ;

a transform definition developed as a logic component of an application program, the transform definition configured to generate a transformed value of the data item property for association with an element property of the user interface element , wherein the transform definition is interjected in a data path between the data item and the user interface, wherein the transform definition changes how the data item value is represented ~~displayed in the user interface,~~ wherein representation of the data item is dependent upon the transform definition, wherein the transform definition is separate and distinct from the data style definition and is applied to the data item prior to the presentation ~~characteristics are not integrated with the data item and are integrated with the data style definition,~~ and wherein the transformed value of the data item property being generated from the associated value of the data item property such that the associated value of the data item property is maintained unchanged in a data item database; and

a content presenter configured to apply the data style definition to an instantiation of a display element on the user interface to display one or more of the data items according to the defined visual representation by combining the data from the data items and the information from the data style definition.

2. (Canceled)

3. (Canceled)

4. (Previously Presented) A data programming model as recited in claim 1, wherein the binding definition facilitates one-time binding wherein a user interface property is initialized from a data item property and wherein the user interface property does not update when changes are made to the data item property after the initialization.

5. (Currently Amended) A data programming model as recited in claim 1, ~~further comprising a~~ wherein the transform definition is configured to generate a transformed value of the data item property for compatible association with the element property of the user interface element.

6. (Original) A data programming model as recited in claim 1, further comprising an update logic component configured to receive a data item update that corresponds to a change of the associated value of the data item property, and further configured to initiate that the binding definition update the element property of the user interface element with the associated value change of the data item property.

7. (Original) A data programming model as recited in claim 1, further comprising an update logic component configured to receive a user interface element update that corresponds to a change of the value of the element property of the user interface element, and further configured to update the associated data item property with the value change of the element property of the user interface element.

8. (Original) A data programming model as recited in claim 1, further comprising:

a data context property configured to define the data item as the data source of the user interface element; and

an additional binding definition configured to associate an element property of an additional user interface element with an additional data item property of the data item, the additional user interface element having a

dependent association to the user interface element, and the additional binding definition further configured to default to the data context property to define the data item as the data source of the additional user interface element.

9. (Original) A data programming model as recited in claim 1, further comprising:

a data context property configured to define the data item as the data source of the user interface element;

an additional binding definition configured to associate an element property of an additional user interface element with an additional data item property of the data item, the additional user interface element having a dependent association to the user interface element, and the additional binding definition further configured to default to the data context property to define the data item as the data source of the additional user interface element; and

wherein a change of the value of the element property of the user interface element initiates a change of a value of the element property of the additional user interface element according to the default data context property.

10. (Original) A data programming model as recited in claim 1, further comprising:

a collection of data items; and

a representation of the data items each configured for display in a user interface display element that is associated with a referenced data item in the representation of the data items.

11. (Original) A data programming model as recited in claim 1, further comprising:

a collection of data items; and

a first representation of the data items in the collection and at least a second different representation of the data items in the collection, the first representation and the second different representation each being configured to reference the data items in the collection.

12. (Canceled)

13. (Currently Amended) A data programming model as recited in claim 1, wherein the data style definition further comprises an internal representation of the user interface element, wherein the user interface element represents the combination of the value of the data item property and information from the data style definition, the data style definition describing how to display the data item property;

a template, wherein the template is a general form of the visual representation of the data item, wherein the template requires that part of the visual representation be obtained as a data item property value; and

a tree assembler module comprising a style lookup module and ~~a~~ the data bind module, wherein the data bind module locates any style elements of the data item and binds a property of a user interface element to a data item property.

14. (Original) A data programming model as recited in claim 1, further comprising a data style definition configured to define a visual representation of data items as a data tree.

15. (Canceled)

16. (Canceled)

17. (Currently Amended) A computing system, comprising:
one or more processing units;
a system memory coupled to the processing unit, storing at least one of a plurality of programs, the programs comprising:

an application program configured to generate a user interface having a display element to display a representation of a data item;

a data programming model configured to implement a binding definition to associate a display element property of the display element with a data item property of the data item such that a value of the data item property is displayed as the representation of the data item, wherein the data programming model comprises:

a transform definition developed as a logic component of the application program, the transform definition configured to generate a transformed value of the data item property for association with an element property of the user interface element by a binding definition, wherein the transform definition is interjected in a data path between the data item and the user interface, wherein the transform definition changes how the data item value is represented ~~displayed in the user interface~~, and wherein representation of the data item is dependent upon the transform definition; and

a data style definition configured to define the representation of the data item on the user interface, wherein the associated value of the data item property influences the appearance of a feature of the user interface element, wherein data items are maintained independently from the data style definition, and wherein the data item is represented as a subtree of

the user interface element and data item properties are represented as part of the subtree, and wherein the data style definition is separate and distinct from the transform definition and is applied to the data item after the transform definition, wherein the data style definition further includes

an internal representation of the user interface element, wherein the user interface element represents the combination of the value of the data item property and information from the data style definition, the data style definition describing how to display the data item property;

a template, wherein the template is a general form of the visual representation of the data item, wherein the template requires that part of the visual representation be obtained as a data item property value; and

a tree assembler module comprising a style lookup module and a data bind module, wherein the data bind module locates any style elements of the data item and binds a property of a user interface element to a data item property.

18. (Previously Presented) A computing system as recited in claim 17, wherein the data programming model comprises a transform definition

to generate a transformed value of the data item property for association with the display element property.

19. (Previously Presented) A computing system as recited in claim 17, wherein the data programming model comprises a transform definition to generate a transformed value of the data item property for association with the display element property, the transformed value of the data item property being generated from the value of the data item property such that the value of the data item property is maintained unchanged in a data item database.

20. (Previously Presented) A computing system as recited in claim 17, wherein the data programming model comprises a transform definition developed as a logic component of the application program, the transform definition configured to generate a transformed value of the data item property for association with the display element property.

21. (Previously Presented) A computing system as recited in claim 17, wherein the data programming model comprises a transform definition configured to generate a transformed value of the data item property for compatible association with the display element property.

22. (Previously Presented) A computing system as recited in claim 17, wherein the data programming model comprises an update logic component configured to receive a data item update that corresponds to a change of the value of the data item property, and the data programming model is further configured to initiate that the binding definition update the display element property with the value change of the data item property.

23. (Previously Presented) A computing system as recited in claim 17, wherein the data programming model comprises an update logic component configured to receive a display element update that corresponds to a change of the value of the display element property, and is further configured to update the associated data item property with the value change of the display element property.

24. (Previously Presented) A computing system as recited in claim 17, wherein the data programming model comprises:

a data context property configured to define the data item as the data source of the display element; and

an additional binding definition configured to associate a display element property of an additional display element with an additional data item property of the data item, the additional display element having a dependent association to

the display element, and the additional binding definition further configured to default to the data context property to define the data item as the data source of the additional display element.

25. (Original) A computing system as recited in claim 17, further comprising a collection of data items, and wherein the data programming model includes a representation of the data items each configured for display in a user interface display element that is associated with a referenced data item in the representation of the data items.

26. (Previously Presented) A computing system as recited in claim 17, further comprising a collection of data items, and wherein the data programming model comprises a first representation of the data items in the collection and at least a second different representation of the data items in the collection, the first representation and the second different representation each being configured to reference the data items in the collection.

27. (Canceled)

28. (Previously Presented) A computing system as recited in claim 17, wherein the data style definition is configured to define a template for the display element to display the value of the data item property.

29. – 31. (Canceled)

32. (Currently Amended) A method, comprising:
developing a data item having a data item property with an associated value for display on a user interface generated by an application program, the data item being developed independently of display-related information corresponding to the user interface;

developing the application program independently of the data item;

defining a binding association between an element property of a user interface element and the data item property such that a value of the element property is defined by the association to the data item property;

developing a transform definition as a logic component of the application program, the transform definition configured to generate a transformed value of the data item property for association with an element property of the user interface element by a binding definition, wherein the transform definition is interjected in a data path between the data item and the user interface, wherein the transform definition changes how the data item value is represented

~~displayed in the user interface, and wherein representation of the data item is dependent upon the transform definition, wherein the presentation characteristics are not integrated with the data item;~~

developing a data style definition configured to define the presentation of the data item on the user interface, wherein the associated value of the data item property influences the appearance of a feature of the user interface element, wherein the data item is maintained independently from the data style definition, and wherein the data item is represented as a subtree of the user interface element and the data item property is represented as part of the subtree, and wherein the data style definition is separate and distinct from the transform definition and is applied to the data item after the transform definition;

developing a template, wherein the template is a general form of the visual representation of the data item, wherein the template requires that part of the visual representation be obtained as a data item property value; and

using a tree assembler module comprising a style lookup module and a the data bind module, wherein the data bind module locates any style elements of the data item and binds a property of a user interface element to a data item property.

33. (Original) A method as recited in claim 32, further comprising generating a transformed value of the data item property for association with the element property of the user interface element.

34. (Original) A method as recited in claim 32, further comprising generating a transformed value of the data item property for association with the element property of the user interface element, the transformed value of the data item property being generated from the associated value of the data item property such that the associated value of the data item property is maintained unchanged in a data item database.

35. (Original) A method as recited in claim 32, further comprising developing a transform definition as a logic component of the application program, the transform definition being configured to generate a transformed value of the data item property for association with the element property of the user interface element.

36. (Original) A method as recited in claim 32, further comprising generating a transformed value of the data item property for compatible association with the element property of the user interface element.

37. (Original) A method as recited in claim 32, further comprising receiving a data item update that corresponds to a change of the associated value of the data item property, and initiating that the binding definition update the element property of the user interface element with the associated value change of the data item property.

38. (Original) A method as recited in claim 32, further comprising receiving a user interface element update that corresponds to a change of the value of the element property of the user interface element, and updating the associated data item property with the value change of the element property of the user interface element.

39. (Original) A method as recited in claim 32, further comprising:
defining a data context property to identify the data item as the data source of the user interface element; and

defining an additional binding association between an element property of an additional user interface element and an additional data item property of the data item, the additional user interface element having a dependent association to the user interface element, and the additional binding association defaulting to the data context property to identify the data item as the data source of the additional user interface element.

40. (Original) A method as recited in claim 32, further comprising:
defining a collection of data items; and
developing a representation of the data items for display in user interface display elements that are each associated with a referenced data item in the representation of the data items.

41. (Original) A method as recited in claim 32, further comprising:
defining a collection of data items; and
developing a first representation of the data items in the collection and at least a second different representation of the data items in the collection, the first representation and the second different representation each referencing the data items in the collection.

42. (Canceled)

43. (Previously Presented) A method as recited in claim 32, wherein the data style definition further comprises an internal representation of the user interface element, wherein the user interface element represents the combination of the value of the data item property and information from the

data style definition, the data style definition describing how to display the data item property.

44. (Previously Presented) A method as recited in claim 32, wherein the data style definition comprises a data tree.

45. (Previously Presented) A method as recited in claim 44, wherein the data items are maintained independently from the data style definition.

46. (Previously Presented) A method as recited in claim 32, further comprising:

applying the data style definition to an instantiation of a display element on the user interface to display one or more of the data items according to the defined visual representation.

47. (Previously Presented) A method as recited in claim 32, further comprising declaring an instance of a data class which corresponds to a type of data as a resource, and wherein defining the binding association comprises referring to the data class in a declaration of the binding association.